อนันต์ กิติบรรลุ

เมาส์ยูทิลิตี้
กุญแจสู่โปรแกรม

ยูทิลิตี้ที่พัฒนาแล้ว เพื่อปรับโปรแกรม ให้ทำงานกับเมาส์ได้

เราคงไม่อาจปฏิเสธได้ว่า เมาส์จัดเป็นอุปกรณ์นำเข้าอีกอย่างหนึ่งที่มีความสำคัญมาก รองจากคีย์บอร์ดอันเป็นอุปกรณ์นำเข้ามาตรฐานที่ใช้กันทั่ว ๆ ไปในคอมพิวเตอร์ โดยเฉพาะอย่างยิ่งในการนำคอมพิวเตอร์มาใช้งานทางด้านกราฟิก ดังจะเห็นได้จากการใช้งานบนเครื่องแมคอินทอชหรือเครื่องประเภทเวอร์กสเตชันทั้งหลาย หรือแม้แต่เครื่องไมโครคอมพิวเตอร์ IBM/PC คอมแพต-ทิเบิล ทั่ว ๆ ไป เช่น การใช้งานบนไมโครซอฟต์วินโดว์จะเห็นว่าหากไม่มีเมาส์แล้ว การใช้งานค่อนข้างยุ่งยาก แต่การจะนำโปรแกรมที่เราพัฒนากันขึ้นมาให้สามารถใช้เมาส์ได้ แม้จะไม่ใช่เรื่องยากลำบากนัก แต่ก็ทำให้ผู้เขียนโปรแกรมต้องมาเสียเวลาศึกษาอะไรอีกหลายอย่าง ดังนั้นผมจึงเกิดความคิดในการสร้างยูทิลิตี้ขึ้นมาใช้งานเอง เพื่อให้ทำงานได้สะดวกและใช้ง่าย โดยไม่จำเป็นที่ผู้ใช้งานหรือโปรแกรมเมอร์ต้องมาจัดการเกี่ยวกับการทำงานภายในเลย เพียงแต่เรียกฟังก์ชันที่กำหนดไว้ก็เพียงพอแล้ว

ลักษณะการทำงานของ utility kit นี้

ลักษณะของยูทิลิตี้นี้ถูกออกแบบให้มีความสามารถคล้ายกับการทำงานของคีย์บอร์ดในระบบ DOS คือตัว BIOS จะเป็นตัวจัดการนำ scan code

```
typedef struct MouseStruc {  /* Message Structure */
        int class; /* ชนิดของเมาส์ */
        int x; /* ตำแหน่งที่เกิด */
        int y;
} InputMouseStruc, *InputMouse;
```

รูปที่ 1 โครงสร้างข้อมูลของ utility kit

และ ASCII code ที่ผู้ใช้กดไปบนแป้นพิมพ์ใส่ไว้ในคีย์บอร์ดบัฟเฟอร์ขนาด 16 word โดยจัดเก็บแบบ Circular queue และถ้าหากค่าในคีย์บอร์ดบัฟเฟอร์เต็มเครื่องจะส่งเสียงบี๊พออกมา แต่ในยูทิลิตี้ตัวนี้ได้ถูกออกแบบให้มีโครงสร้างข้อมูลดังรูปที่ 1

รายละเอียดภายในโครงสร้างข้อมูล

– Class เป็นชนิดของเมาส์แมสเสจแต่ละอันได้แก่

IE_LBUTTONUP เมื่อปล่อยเมาส์ปุ่มซ้าย

IE_LBUTTONDOWN เมื่อกดเมาส์ปุ่มซ้าย

IE_LDBLCLICK เมื่อกดเมาส์ปุ่มซ้ายสองครั้งติดต่อกัน (เหมือนที่ใช้ในไมโครซอฟต์วินโดว์ เพื่อเลือกแอพพลิเคชันมาทำงาน)

IE_MBUTTONUP เมื่อปล่อยเมาส์ปุ่มกลาง

IE_MBUTTONDOWN เมื่อกดเมาส์ปุ่มกลาง

IE_MDBLCLICK เมื่อกดเมาส์ปุ่มกลางสองครั้งติดต่อกัน

IE_RBUTTONUP เมื่อปล่อยเมาส์ปุ่มขวา

IE_RBUTTONDOWN เมื่อกดเมาส์ปุ่มขวา

IE_RDBLCLICK เมื่อกดเมาส์ปุ่มขวาสองครั้งติดต่อกัน

IE_MOUSEMOVE เมื่อเคลื่อนเมาส์

– ตัวแปร x และ y เป็นตัวบอกตำแหน่งของการเกิดเมาส์แมสเสจว่าอยู่ตำแหน่งใดบนจอภาพโดย

x แสดงถึง ตำแหน่งในแนว 'แถว'

y แสดงถึง ตำแหน่งในแนว 'คอลัมน์'

จะเห็นว่าค่าเมาส์แมสเสจแต่ละตัวจะมีขนาด 3 ไบต์ และสำหรับค่าของเมาส์บัฟเฟอร์นั้นจะมีขนาดตามที่ได้กำหนดไว้ใน mseutil.h คือ

```
# define MAXMESSAGE 100
/* mouse circular queue buffer */
```

ซึ่งจะเป็นค่าบัฟเฟอร์ของเมาส์ที่สามารถเปลี่ยนได้หากผู้ใช้คิดว่ามากไปหรือน้อยไป เราสามารถสังเกตได้ว่าหากแมสเสจเมาส์เต็มจนไม่สามารถที่จะนำไปไว้ในเมาส์บัฟเฟอร์ได้ มันจะส่งเสียงบี๊พออกมาเช่นเดียวกับเมื่อคีย์บอร์ดบัฟเฟอร์เต็ม โดยเรียกฟังก์ชันของ BIOS (เพื่อไม่ให้เกิดปัญหาเรื่อง reentrance ของ DOS เพราะเกิดในช่วงอินเตอร์รัพต์ คือ int 10h ฟังก์ชัน 14) โดยการส่งค่า bell character ('\700') ซึ่งจะทำให้เกิดเสียงบี๊พขึ้น

การนำ Mouse utility ไปใช้งาน

1. การเรียกฟังก์ชันโดยดูคำอธิบายในตารางที่ 1

```
InitMouse (int startx, int starty, int
           maxx, int maxy, Mode
           screen mode);
```

return value : ถ้าเป็น 0 หรือ FALSE คือทำงานไม่สำเร็จ อาจเป็นเพราะยังไม่ install mouse driver แต่ถ้าเป็น 1 หรือ TRUE ทำงานสำเร็จหรือติดต่อกับเมาส์ได้

2. คอยวนลูปเพื่อเรียกฟังก์ชัน

```
{InputMouse GetMouse(void);}
```

return value : InputMouse ซึ่งเป็นพอยน์เตอร์ที่ชี้ไปยังตำแหน่งเมาส์บัฟเฟอร์

parameter	คำอธิบาย
startx	ตำแหน่งที่จะให้เมาส์พอยน์เตอร์ไปปรากฏบนจอภาพในแนว 'แถว'
starty	ตำแหน่งที่จะให้เมาส์พอยน์เตอร์ไปปรากฏบนจอภาพในแนว 'คอลัมน์'
maxx	ค่าสูงสุดที่ให้ mouse เคลื่อนที่ไปได้ในแนว 'แถว'
maxy	ค่าสูงสุดที่ให้ mouse เคลื่อนที่ไปได้ในแนว 'คอลัมน์'
mode	มี 2 ค่าคือ GRAPHIC_MODE ใช้ในโหมดกราฟิก TEXT_MODE ใช้ในโหมดตัวอักษร ทั้ง 2 โหมดฟังก์ชันนี้จะแสดงเมาส์พอยน์เตอร์ให้

ตารางที่ 1 อธิบายพารามิเตอร์ของฟังก์ชัน InitMouse

3. เรียกฟังก์ชัน CloseMouse (void); เพื่อเลิกการใช้เมาส์
return value : ไม่มี

รายละเอียดประกอบการใช้งานใน mouse utility

– draw_pointer(void) เพื่อให้เมาส์พอยน์เตอร์ปรากฏบนจอในโหมดกราฟิก และใช้ show_mouse_cursor (void) ในโหมดตัวอักษร

– erase_pointer(void) เพื่อให้เมาส์พอยน์เตอร์ถูกซ่อนเอาไว้ไม่ให้ปรากฏบนจอในโหมดกราฟิก และใช้ hide_mouse_cursor(void) สำหรับโหมดตัวอักษรควรเรียกทุกครั้งเมื่อจะแสดงข้อความหรืออะไรก็ตาม เพื่อไม่ให้พอยน์เตอร์มามีผลต่อการแสดงผล หลังจากนั้นจึงค่อยเรียก draw_pointer หรือ show_mouse_cursor เพื่อแสดงเมาส์พอยน์เตอร์

– SetDblClickRate(int rate) ใช้เพื่อเซตเวลาหรืออัตราเร็วในการ double click บนเมาส์ ตามเวลาที่กำหนดโดยค่า rate (เวลาที่กำหนดไว้คือ 10 หน่วย ซึ่งเป็นเวลาของ BIOS time 1 หน่วย คิดเป็น 55 มิลลิวินาที) สำหรับเวลาที่ใช้ในยูทิลิตี้นี้ได้กำหนดไว้เป็น 8 หน่วย

– void SendInputMouse(class, arg1,arg2) ฟังก์ชันนี้มีความสำคัญมาก เพราะถูกเรียกใช้จากอีเวนต์เมาส์ ทุกครั้งที่เกิดอีเวนต์ขึ้นเพื่อจัดสรรเมาส์แมสเสจลงคิว โดยจะใส่แมสเสจที่เกิดขึ้นลงสู่ท้ายคิว อนึ่งลักษณะของคิวจะมีโครงสร้างเป็นเซอร์คูลาร์คิว คือเมื่อแมสเสจมาถึงท้ายคิวมันจะ ship ไปที่หัวคิวในการรับแมสเสจต่อไป (ดูรูปที่ 2)

– InputMouse GetMouse( ) ฟังก์ชันนี้เป็นส่วนทั่วไปที่นำไปใช้เรียก โดยมันจะทำการดึงเมาส์แมสเสจออกจากคิว โดยจะดึงจากหัวคิวเสมอ เมื่อไม่มีอะไรอยู่ในคิวเลยมันจะรอคอยจนกว่ามีเมาส์แมสเสจเข้ามาเสียก่อน (ดูรูปที่ 2)

– void int InitMouse(..) ส่วนนี้จะมีการเรียกฟังก์ชัน set_mouse_function (int mask, void interrupt (*proc) ( )) ซึ่งเป็นฟังก์ชันที่จะบอกว่าเมื่อเกิดเมาส์อินเตอร์รัพต์ขึ้นให้เรียกระยะไกลมาที่ส่วน 'proc' โดยการเกิดนั้น set โดยค่า mask ในที่นี้เราให้เกิดทุกกรณีเลยคือ การเคลื่อนย้ายการกดและการปล่อยเมาส์ทุกปุ่ม (ดูรูปที่ 3)

– void interrupt EventMouse( ) ฟังก์ชันนี้ถือเป็นหัวใจในการติดต่อกับเมาส์ โดยจะเป็นส่วนที่ดึงการทำงานของเมาส์มา หรือเรียกอีกอย่างว่า การดักอินเตอร์รัพต์ส่วนของเมาส์มาก็ได้ ภายในนี้จะทำการส่งแมสเสจที่เกี่ยวกับเมาส์ทั้งหมด ได้แก่ การกดปุ่มซ้าย กดปุ่มขวา การปล่อยปุ่มกลาง เป็นต้น และจะมีส่วนของการ check double click ซึ่งในส่วนของเมาส์ไดรเวอร์เองยังไม่มีฟังก์ชันที่ตรวจส่วนนี้ให้จึงต้องทำเอง โดยการตรวจเวลาของ BIOS time คือ ส่วนแอดเดรส 40:6C ขนาด 4 ไบต์ ว่าในการเกิดแมสเสจนี้ในครั้งก่อนเกิดตอนไหน แล้วจึงตรวจว่าแมสเสจที่เข้ามาตอนนี้เกิดเมื่อใด จึงค่อยตรวจสอบความแตกต่างของเวลาทั้งสอง ว่าอยู่ในช่วงที่กำหนดหรือไม่ ถ้าใช่ก็ส่งเป็น message double click แทน แต่ถ้าไม่ใช่ก็ส่งเป็น message click down ธรรมดาแทนแล้วแต่ว่าเป็นปุ่มใด

หมายเหตุ ฟังก์ชันนี้จริง ๆ แล้วเกิดจากการเรียกระยะไกลของ interrupt 33h ไม่ใช่เป็นอินเตอร์รัพต์รูทีน แต่อินเตอร์รัพต์จริง ๆ นั้นเกิดที่ interrupt 33h แล้วจึงเรียกระยะไกลมาที่ฟังก์ชันนี้ ดังนั้นดาต้าเซกเมนต์จะอยู่ที่ตัว interrupt 33h ไม่ได้อยู่ที่ตัวโปรแกรม นั่นคือการที่จะสามารถอ้างดาต้าเซกเมนต์ของโปรแกรมเพื่ออ้างถึงข้อมูลหรือตัวแปรจะต้องเขียนส่วนนี้เป็นอินเตอร์รัพต์ เพราะตัวเทอร์โบซีจะจัดการแปลงรหัสโดยโอนดาต้าเซกเมนต์มาเป็นตัวของโปรแกรมจริง และในตอนเสร็จสิ้นโปรแกรมก็ออกไปตามปกติไม่ได้เพราะการออกโดยปกติจะทำคำสั่ง IRET ซึ่งจะดึง return address 2 word และ popf อีก 1 word ออกมาจากสแต็ก แต่เนื่องจากการเรียกระยะไกลมานั้นมีการใส่ค่าในสแต็กเพียง 2 word คือ return address ดังนั้นจึงต้อง force คำสั่ง _emit_ ลงไปเพื่อให้เลิกโดยทำคำสั่ง RETF นั่นเอง (รายละเอียดขอให้ดูจากรูปที่ 4 ส่วนฟังก์ชันนี้ได้)

– draw_pointer( ) และ erase_pointer( ) เป็นฟังก์ชันที่สร้างขึ้นเองในการวาดเมาส์พอยน์เตอร์ โดยใช้ฟังก์ชัน putimage( ) ในการวาดพอยน์เตอร์และ getimage( ) ในการเก็บฉากหลังที่ไม่ใช้ฟังก์ชันของตัวเมาส์ไดรเวอร์ เพราะไดรเวอร์บางตัวไม่สนับสนุนฟังก์ชันนี้ แต่สำหรับในโหมดตัวอักษรสามารถแสดงให้ตามปกติ

สำหรับฟังก์ชันอื่น ๆ ขอให้ดูจากรูปที่ 5 และ 6 ได้ และตัวอย่างการใช้งานที่ทำไว้มี 2 ชุดด้วยกัน มี textmse.c ซึ่งเป็นการนำ mouse utility tools นี้ไปใช้ในโหมดตัวอักษร ส่วนในโหมดกราฟิกนั้นให้ดูใน graphmse.c ท้ายนี้ผมหวังว่า mouse utility tools kit ชุดนี้ การที่จะไปใช้สร้างแอพพลิเคชันที่ดี ๆ แก่วงการคอมพิวเตอร์ในบ้านเราต่อไปนะครับ..สวัสดี!

รูปที่ 2
โปรแกรม MSEUTIL.H

```
/**
     FILE NAME : MSEUTIL.H
     Programmer: Anan Kittibunlu
     Create:   11/Feb/1991
     Update:   16/March/1991
     Purpose:  Prototype function for mouse utility (mseutil.c)
     Comment:  This utility tools can use only TURBO C Compiler

     Copyright (C) 1991 Chulalongkorn University
**/

#ifndef _MOUSE_H
#define _MOUSE_H

typedef struct {
        int class;    /* class of mouse */
        int x;        /* pointer x */
        int y;        /* pointer y */
}InputMouseStruc, *InputMouse;

/* define bit mask of class of Message */
#define IE_LBUTTONUP      0X0001
#define IE_LBUTTONDOWN    0X0002
#define IE_LDBLCLICK      0X0003
#define IE_MBUTTONUP      0X0004
#define IE_MBUTTONDOWN    0X0005
#define IE_MDBLCLICK      0X0006
#define IE_RBUTTONUP      0X0007
#define IE_RBUTTONDOWN    0X0008
#define IE_RDBLCLICK      0X0009
#define IE_MOUSEMOVE      0X000A

#define MAXMESSAGE 100      /* mouse circular queue buffer */
#define T_DBLCLICK 8        /* double click rate default */

#ifndef TRUE
#define TRUE   1
#define FALSE  !TRUE
#endif

typedef enum {       /* mode screen for mouse pointer */
   GRAPHIC_MODE,
   TEXT_MODE
}ModeScreen;

/****************** Utility Fuction to use in your program **********/
void  draw_pointer(void);
void  erase_pointer(void);
InputMouse GetMouse(void);
void SetDblClickRate(int rate);
int InitMouse(int startx, int starty, int maxx, int maxy, ModeScreen mode)
void  CloseMouse(void);

/****************** prototype of mouse function **********************/
/*****************  for you to call other mouse function  ************/
/*****************  these functions are in mouse.c  ******************/
int     mouse_reset(void);

void    show_mouse_cursor(void);
void    hide_mouse_cursor(void);

void    get_mouse_position(int *button, int *hor, int *ver);
void    set_mouse_position(int hor, int ver);

int     get_mouse_button_presses(int *button, int *hor, int *ver);
int     get_mouse_button_releases(int *button, int *hor, int *ver);
void    read_mouse_motion(int *hor, int *ver);

void    set_mouse_horizontal_limit(int min, int max);
void    set_mouse_vertical_limit(int min, int max);

void    set_mouse_graphics_cursor(int hor, int ver, void far *mark);
void    set_mouse_text_cursor(int cursor_type, int start, int stop);

void    mouse_lightpen_on(void);
void    mouse_lightpen_off(void);

void    mouse_condition_off(int x1, int y1, int x2, int y2);

void    set_mouse_pixel_ratio(int hor, int ver);
void    set_mouse_double_threshold(int threshold);

/*  trap interrupt mouse driver */
void    set_mouse_function(int mask, void interrupt (*proc) ());
/*************************************************************************

#endif
```

รูปที่ 3 โปรแกรม MSEUTIL.C

```
/**
     FILE NAME : MSEUTIL.C
     Programmer: Anan Kittibunlu
     Create:   11/Feb/1991
     Update:   16/March/1991
     Purpose:  For interface with mouse driver and create circular queue
               for keep message mouse (look like circular queue keyboard buffer)
     Comment:  This utility tools can use only TURBO C Compiler

     Copyright (C) 1991 Chulalongkorn University
**/

#include <stdio.h>
#include <dos.h>
#include <graphics.h>
#include <alloc.h>
#include "mseutil.h"

#define Word4  unsigned long

/* ======= internal routine ============================== */
static void interrupt EventMouse (void);
static void SendInputMouse(int, int, int);

InputMouseStruc msgbuf[MAXMESSAGE];     /*  buffer of message */
void interrupt (*Event_Ptr)() ;

static int msgheadPtr = -1;    /* -1 mean empty queue */
static int msgtailPtr = 0;
static unsigned long far *btime = MK_FP(0X40,0X6C);

#define MS_SIZE 7
static int x,y,maxX,maxY;
static char *ms_back;    /* background of mouse pointer */
static screenMode;
static timeDbl = T_DBLCLICK;  /* default */

void    draw_pointer()
{
        int oldc=getcolor();

        setcolor(YELLOW);
        if (x < MS_SIZE) x = MS_SIZE;
        if (y < MS_SIZE) y = MS_SIZE;
        if (x+MS_SIZE > maxX) x = maxX-MS_SIZE;
        if (y+MS_SIZE > maxY) y = maxY-MS_SIZE;
        getimage(x-MS_SIZE,y-MS_SIZE,x+MS_SIZE,y+MS_SIZE,ms_back);
        line(x-MS_SIZE,y-MS_SIZE,x+MS_SIZE,y+MS_SIZE);
        line(x-MS_SIZE,y+MS_SIZE,x+MS_SIZE,y-MS_SIZE);
        setcolor(oldc);
}

void    erase_pointer()
{
        if (x < MS_SIZE) x = MS_SIZE;
        if (y < MS_SIZE) y = MS_SIZE;
        if (x+MS_SIZE > maxX) x = maxX-MS_SIZE;
        if (y+MS_SIZE > maxY) y = maxY-MS_SIZE;
        putimage(x-MS_SIZE,y-MS_SIZE,ms_back,COPY_PUT);
}

static void put(char ch)
{
        _AH=0x0E;
        _AL=ch;
        geninterrupt(0x10);
}

void SendInputMouse(class,arg1,arg2)
unsigned char class;
int arg1,arg2;
/*
     take message from mouse to keep in input message queue
     by filled in next priviouse message queue
 */
{

        InputMouse p;
        register int t;

        if (msgheadPtr >= 0)  {    /* Queue Not Empty */
            register int n;
            n = msgtailPtr - msgheadPtr;
            if (n < 0) n = -n;
                t = msgtailPtr + 1;
                if (t > (MAXMESSAGE-1)) t = 0;
                if (t == msgheadPtr) {    /* full? */
                        put (7); /* beep */
                        return;
                }
                p = msgbuf + t;
                msgtailPtr = t;
                goto giveValue;
        }                       /* Queue Empty */
        p = msgbuf;  /* point to head queue */
        if (msgheadPtr < 0) msgheadPtr = msgtailPtr = 0;
giveValue:
        p->class = class;
        p->x =  arg1;
        p->y =  arg2;
}


InputMouse GetMouse()
/**
    get message from header of queue
    input : none

    output : InputMessage

**/
{
        int h;
        while (msgheadPtr < 0);  /* wait until have message */
        h = msgheadPtr;
        if (++msgheadPtr > (MAXMESSAGE - 1)) msgheadPtr = 0;
        if (h == msgtailPtr) msgheadPtr = -1;
        return  msgbuf+ h;
};

void SetDblClickRate(int rate)
/**
```

รูปที่ 3 (ต่อ)

```
   set double click time rate for application
   rate is 1/18.2 sec unit
**/
{
        if (rate < 0 || rate >20) return;
        timeDbl = rate;
}

void interrupt EventMouse()
/**
   Mouse Event Handler
   this routine hooked mouse event when mouse was done
**/
{
/* define mask bit when int 33h call this event */
#define MMOVE   1
#define LDOWN   2
#define LUP     4
#define RDOWN   8
#define RUP    16
#define MDOWN  32
#define MUP    64
/* define check dblclick if move mouse more than this */
/* and occurs dblclick then don't care                */
#define MOVEX  20
#define MOVEY  20

        extern int abs(int);
        unsigned int sax, scx, sdx ;   /* flag int 33 call far     */
        static int firstLT = TRUE;
        static int firstMT = TRUE;
        static int firstRT = TRUE;
        static unsigned int oldLX, oldLY;
        static unsigned int oldMX, oldMY;
        static unsigned int oldRX, oldRY;
        static unsigned long oldLTime = 0; /* time for check left Dbl click   */
        static int oldLClick = 0;            /* flag for check left dbl-click   */
        static unsigned long oldMTime = 0; /* time for check middle Dbl click */
        static int oldMClick = 0;                /* flag for check middle dbl-click
        static unsigned long oldRTime = 0; /* time for check right Dbl click  */
        static int oldRClick = 0;                /* flag for check right dbl-click

        disable();
        sax = _AX;      /* call mask bit */
        scx = _CX;      /* horizontal location */
        sdx = _DX;      /* vertical location */
        enable();

        if (sax & MMOVE) {
            if (screenMode == GRAPHIC_MODE)  {
                erase_pointer();
                x = scx; y = sdx;    /* update x & y */
                draw pointer();
            }
            SendInputMouse(IE_MOUSEMOVE, scx, sdx);
        }
        if (sax != MMOVE) {  /* yes return */
          if(sax & LDOWN) {
                if (firstLT) {
                        oldMX = scx; oldMY = sdx;  /* save pointer */
                        firstMT = FALSE;
        }
        if(abs(scx-oldLX) > MOVEX ||
                abs(sdx-oldLY) > MOVEY) {
                oldLClick = FALSE; /* don't check dbl click */
        }
        oldLX = scx; oldLY = sdx; /* change to current pointer */
         if (oldLClick) /* second occur */
                if (*btime-oldLTime < timeDbl) {  /* dbl-click */
                        SendInputMouse(IE_LDBLCLICK, scx, sdx);
                        oldLClick = FALSE; /* clear */
                }
                else { /* too long for dbl-click */
                        SendInputMouse(IE_LBUTTONDOWN, scx, sdx);
                        oldLTime = (Word4) *btime; /* update time */
                }
         else { /* fist occur */
                SendInputMouse(IE_LBUTTONDOWN, scx, sdx);
                oldLTime = (Word4) *btime;
                oldLClick++;  /* to check dbl click for next event */
         }
         oldRClick = oldMClick = FALSE; /* reset flag */
}  /* LDOWN */

if (sax & LUP) {
        SendInputMouse(IE_LBUTTONUP, scx, sdx);
}

if (sax & RDOWN) {
        if (firstRT) {
                oldRX = scx; oldRY = sdx;  /* save pointer */
                firstRT = FALSE;
        }
        if (abs(scx-oldRX) > MOVEX ||
                abs(sdx-oldRY) > MOVEY) {
                oldRClick = FALSE;
        }
        oldRX = scx; oldRY = sdx; /* change to current pointer */
        if (oldRClick) /* second occur */
                if (*btime-oldRTime < timeDbl) {  /* dbl-click */
                   SendInputMouse(IE_RDBLCLICK, scx, sdx);
                   oldRClick = FALSE; /* clear */
        }
        else { /* too long for dbl-click */
                SendInputMouse(IE_RBUTTONDOWN, scx, sdx);
                oldRTime = (Word4) *btime; /* update time */
        }
        else { /* fist occur */
                SendInputMouse(IE_RBUTTONDOWN, scx, sdx);
                oldRTime = (Word4) *btime;
                oldRClick++;
        }
        oldLClick = oldMClick = FALSE; /* reset flag */
}  /* RDOWN */

if (sax & RUP) {
        SendInputMouse(IE_RBUTTONUP, scx, sdx);
}

if (sax & MDOWN) {
      if (firstMT) {
```

→

รูปที่ 3 (ต่อ)

```
                oldMX = scx; oldMY = sdx;  /* save pointer */
                firstMT = FALSE;
             }
             if (abs(scx-oldMX) > MOVEX ||
                   abs(sdx-oldMY) > MOVEY) {
                oldMClick = FALSE;
             }
             oldMX = scx; oldMY = sdx; /* change to current pointer */
             if (oldMClick) /* second occur */
                if (*btime-oldMTime < timeDbl) {  /* dbl-click */
                SendInputMouse(IE_MDBLCLICK, scx, sdx);
                oldMClick = FALSE; /* clear */
             }
             else { /* too long for dbl-click */
                SendInputMouse(IE_MBUTTONDOWN,scx,sdx);
                oldMTime = (Word4) *btime; /* update time */
             }
             else { /* fist occur */
                SendInputMouse(IE_MBUTTONDOWN, scx, sdx);
                oldMTime = (Word4) *btime;
                oldMClick++;
             }
             oldLClick = oldRClick = FALSE; /* reset flag */
          }  /* MDOWN */

          if (sax & MUP) {
             SendInputMouse(IE_MBUTTONUP, scx, sdx);
          }
       }
       __emit__(0x89,0xEC);     /* MOV  SP,BP  (restore stack pointer) */
       __emit__(0x5D,           /* POP  BP  */
                0x5F,           /* POP  DI  */
                0x5E,           /* POP  SI  */
                0x1F,           /* POP  DS  */
                0x07,           /* POP  ES  */
                0x5A,           /* POP  DX  */
                0x59,           /* POP  CX  */
                0x5B,           /* POP  BX  */
                0x58,           /* POP  AX  */
                0xCB);          /* RETF     */
}

int InitMouse (int startx, int starty, int maxx, int maxy, ModeScreen mode)
/**
        startx, starty is a location that give pointer to it.
        maxx and maxy is maximum range to mouse moving
**/
{

       if ( !mouse_reset() ) return (FALSE);
       else {
             set_mouse_pixel_ratio(8,8);
             set_mouse_position (x=startx,y=starty);
             set_mouse_horizontal_limit(0,maxX=maxx-1);
             set_mouse_vertical_limit(0,maxY=maxy-1);
             if ((screenMode=mode) == GRAPHIC_MODE) {
                ms_back=(char *)malloc(imagesize(0,0,MS_SIZE*2+1,MS_SIZE*2+1));
                draw_pointer();
             }
             else {
                set_mouse_text_cursor(1,1,7);
```

→

รูปที่ 3 (ต่อ)

```
                show_mouse_cursor();
             }

             disable();
             Event_Ptr = EventMouse;
             set_mouse_function(0x7F, Event_Ptr );
             enable();

             retuin (TRUE);
       }
}

void CloseMouse()
/**
**/
{
       mouse_reset();
       if (screenMode == GRAPHIC_MODE) free(ms_back);
}
```

■

รูปที่ 4 โปรแกรม MOUSE.C

```
/**
     FILE NAME : MOUSE.C
     Programmer: Anan Kittibunlu
     Create:   11/Feb/1991
     Update:   10/March/1991
     Purpose:
               It is functions that call mouse driver(by int 33h)

**/  Copyright (C) 1991 Chulalongkorn University

#include  <dos.h>
#include  "mseutil.h"

int  mouse_reset()
{
     _AX = 0x0000;
     geninterrupt(0x33);
     return((_AX)?(_BX):(_AX));
}

void  show_mouse_cursor()
{
     _AX = 0x0001;
     geninterrupt(0x33);
}

void  hide_mouse_cursor()
{
     _AX = 0x0002;
     geninterrupt(0x33);
}
```

→

รูปที่ 4 (ต่อ)

```
void  get_mouse_position(int *button, int *hor, int *ver)
{
     int  savebx;        /* save Register BX, ES:BX is used to pointer */

     _AX = 0x0003;
     geninterrupt(0x33);
     savebx  = _BX;
     *button = savebx;
     *hor    = _CX;
     *ver    = _DX;
}

void  set_mouse_position(int hor, int ver)
{
     _AX = 0x0004;
     _CX = hor;
     _DX = ver;
     geninterrupt(0x33);
}

int  get_mouse_button_presses(int *button, int *hor, int *ver)
{
     int  saveax, savebx;        /* save Register AX,BX, ES:BX is used to pointer */

     _AX = 0x0005;
     geninterrupt(0x33);
     saveax  = _AX;
     savebx  = _BX;
     *button = savebx;
     *hor    = _CX;
     *ver    = _DX;
     return(saveax);
}

int  get_mouse_button_releases(int *button, int *hor, int *ver)
{
     int   saveax, savebx;       /* save Register AX,BX, ES:BX is used to pointer *

     _AX = 0x0006;
     geninterrupt(0x33);
     saveax  = _AX;
     savebx  = _BX;
     *button = savebx;
     *hor    = _CX;
     *ver    = _DX;
     return(saveax);
}

void  set_mouse_horizontal_limit(int min, int max)
{
     _AX = 0x0007;
     _CX = min;
     _DX = max;
     geninterrupt(0x33);
}

void  set_mouse_vertical_limit(int min, int max)
{
    _AX = 0x0008;
    _CX = min;
    _DX = max;
    geninterrupt(0x33);
}

void  set_mouse_graphics_cursor(int hor, int ver, void far *mark)
{
     _BX = hor;
     _CX = ver;
     _DX = FP_OFF(mark);
     _ES = FP_SEG(mark);   /* _ES use mov ax,[bp+?] , mov ES,ax */
     _AX = 0x0009;         /* _AX set after */
     geninterrupt(0x33);
}

void  set_mouse_text_cursor(int cursor_type, int start, int stop)
{
     _AX = 0x000A;
     _BX = cursor_type;
     _CX = start;
     _DX = stop;
     geninterrupt(0x33);
}

void  read_mouse_motion(int *hor, int *ver)
{
     /* ES:BX is used to pointer */
     _AX = 0x000B;
     geninterrupt(0x33);
     *hor    = _CX;
     *ver    = _DX;
}
void  set_mouse_function(int mask, void interrupt (*proc)() )
{
     _CX = mask;
     _DX = FP_OFF(proc);
     _ES = FP_SEG(proc);
     _AX = 0x000C;
     geninterrupt(0x33);
}


void mouse_lightpen_on()
{
     _AX = 0x000D;
     geninterrupt(0x33);
}

void  mouse_lightpen_off()
{
     _AX = 0x000E;
     geninterrupt(0x33);
}


void  set_mouse_pixel_ratio(int hor, int ver)
{
     _AX = 0x000F;
     _CX = hor;
     _DX = ver;
     geninterrupt(0x33);
}
```

→

รูปที่ 4 (ต่อ)

```
void  set_mouse_double_threshold(int threshold)
{
     _AX = 0x0013;
     _DX = threshold;
     geninterrupt(0x33);
}

void  mouse_condition_off(int x1, int y1, int x2, int y2)
{
     _AX = 0x0010;
     _CX = x1;
     _DX = y1;
     _SI = x2;  /* Turbo C save _SI and _DI automatic */
     _DI = y2;
     geninterrupt(0x33);
}
```

รูปที่ 5 โปรแกรม TEXTMSE.C

```
/**
     FILE NAME : TEXTMSE.C
     Programmer: Anan Kittibunlu
     Create:   16/March/1991
     Purpose:
               Example for use mouse utility (mseutil.c) in Text mode
               When press left button   -> display '+' at cursor
               When press middle button -> display '-' at cursor
               When press right button  -> display '*' at cursor
               to exit program press left double click on
               button of mouse or press ESC key

     Comment:
               This utility tools can use only TURBO C Compiler
               To do this you must ..
               Creat project file "Textmse.prj" that compounded
                textmse.c (mseutil.h)
                mseutil.c (mseutil.h)
                mouse.c   (mseutil.h)

     Copyright (C) 1991 Chulalongkorn University
**/

#include <stdio.h>
#include <bios.h>
#include <conio.h>
#include <stdlib.h>
#include "mseutil.h"
```

รูปที่ 5 (ต่อ)

```
void main ()   /* EXAMPLE USE MOUSE TOOLS */
{
   int forever = 1;
   InputMouse ms_msg;

   clrscr();
   if (!InitMouse(50,50,80*8,25*8,TEXT_MODE)) {
        printf("Error: can't detect mouse \n\
                or Mouse driver is not install\n");
        exit(1);
   }
   while (forever) {

        if(kbhit()) {   /* check have key ? */
           int ch = bioskey(0);

           cprintf("you press key board 0x%x \r\n",ch);
           if(ch == 0x11b) break;  /* exit when press ESC key */

        ms_msg=GetMouse();
        /* have ms_msg */
          switch ( ms_msg->class ) {
                case IE_MOUSEMOVE :
                   break;

                case IE_LBUTTONDOWN:
                    hide_mouse_cursor();
                    gotoxy((ms_msg->x+8)/8,(ms_msg->y+8)/8);
                    putch('+');
                    show_mouse_cursor();
                    break;

                case IE_MBUTTONDOWN:
                    hide_mouse_cursor();
                    gotoxy((ms_msg->x+8)/8,(ms_msg->y+8)/8);
                    putch('-');
                    show_mouse_cursor();
                    break;

                case IE_RBUTTONDOWN:
                    hide_mouse_cursor();
                    gotoxy((ms_msg->x+8)/8,(ms_msg->y+8)/8);
                    putch('*');
                    show_mouse_cursor();
                    break;

                case IE_LDBLCLICK :
                    forever=0;  /* exit */
                    break;
          }
   }
   clrscr();
   printf(" OK. EXIT TO DOS \n");
   CloseMouse();
}
```

รูปที่ 6 โปรแกรม GRAPHMSE.C

```
/**
      FILE NAME : GRAPHMSE.C
      Programmer: Anan Kittibunlu
      Create:   20/March/1991
      Purpose:
                Example for use mouse utility (mseutil.c) in Graphic mod
                When press left button   -> display a little regtangle
                When release left button -> display a line from location
which you press left button
                When release middle button -> display a rectangle from
location which you press middle button
                When press right button  -> clear screen
                to exit program press left double click on
                button of mouse

      Comment:
                This utility tools can use only TURBO C Compiler
                To do this you must ..
                Creat project file "Graphmse.prj" that compounded
                 graphmse.c (mseutil.h)
                 mseutil.c  (mseutil.h)
                 mouse.c    (mseutil.h)

      Copyright (C) 1991 Chulalongkorn University
**/

#include <stdio.h>
#include <graphics.h>
#include <alloc.h>
#include <stdlib.h>
#include <bios.h>
#include "mseutil.h"


void initialsystem(void)
{
        int grdriver,grmode,result;

        detectgraph(&grdriver, &grmode);
        if(grdriver < 0) {
                printf("Error to graphic mode\n");
                printf("Unable to use Graphics Hardware\n");
                exit(1);
        }
        initgraph(&grdriver,&grmode,"");
        if((result = graphresult()) != grOk) {
                printf("Error to graphic mode %s\n",
                        grapherrormsg(result));
                exit(1);
        }
        if (!InitMouse(0,0,getmaxx(),getmaxy(),GRAPHIC_MODE)) {
                closegraph();
                printf("Error: can't detect mouse \n\
                                or Mouse driver is not install\n");
                exit(1);
        }
}


void working_ms_msg(void)
{
int flag=0;
InputMouse ms_msg;


      srand(1000);
      randomize();
      while (!flag) {
         int oldx,oldy;

         ms_msg=GetMouse();
         /* have ms_msg */
         switch ( ms_msg->class ) {
              case IE_MOUSEMOVE :
                   break;

              case IE_LBUTTONDOWN:
                   erase_pointer();
                   bar3d(oldx=ms_msg->x,oldy=ms_msg->y,ms_msg->x+2,ms_msg->y+2,1,1)
                   draw_pointer();
                   break;

              case IE_LBUTTONUP:
                   erase_pointer();
                   bar3d(ms_msg->x,ms_msg->y,ms_msg->x+2,ms_msg->y+2,1,1);
                   line(oldx,oldy,ms_msg->x,ms_msg->y);
                   draw_pointer();
                   break;

              case IE_MBUTTONDOWN:
                   oldx=ms_msg->x; oldy=ms_msg->y;
                   break;

              case IE_MBUTTONUP:
                   erase_pointer();
                   setfillstyle(SLASH_FILL,random(16) /* color */);
                   bar3d(oldx,oldy,ms_msg->x,ms_msg->y,0,1);
                   draw_pointer();
                   break;

              case IE_RBUTTONDOWN:
                   erase_pointer();
                   cleardevice();
                   draw_pointer();
                   break;

              case IE_LDBLCLICK :
                   flag=1;  /* exit */
                   break;
         }
      }
}

void closesystem(void)
{
      CloseMouse();
      closegraph();
}

void main()
{
      initialsystem();
      working_ms_msg();
      closesystem();
```